

Podstawy Programowania

Laboratorium 1

Wprowadzenie do C/C++

prowadzący: mgr inż. Marta Lampasiak, mgr inż. Michał Jaroszczuk

1 Wprowadzenie

Celem zajęć jest poznanie podstaw programowania w języku C/C++, w szczególności ćwiczenia z tworzeniem programów ilustrujących zastosowanie podstawowych instrukcji i konstrukcji programowych języka C/C++: przypisania, rozgałęzienia warunkowego (`if`, `if/else`), wyboru (`switch`, `case`, `break`, `default`). Zagnieżdżanie instrukcji rozgałęziających. Obliczanie wyrażeń matematycznych.

Podstawowy program musi zawsze zawierać funkcję główną `main` (typu `int`). W celu wyświetlenia podstawowego komunikatu można posłużyć się poleceniem `cout` z biblioteki `iostream` lub poleceniem `printf` z `cstdlib`. Biblioteki dołączamy dyrektywą `#include`.

```
1 #include <iostream>
2 #include <cstdlib>
3
4 int main()//komentarz
5 {
6     std::cout << "Wake the f*** up, Samurai. We have city to burn!!\n";
7     printf("Wake the f*** up, Samurai. We have city to burn!!\n");
8
9     return 0;
10 }
```

Należy pamiętać, że używając polecenia `printf` należy zawsze podać odpowiedni typ zmiennej do wyświetlania. Dodatkowo w przypadku liczb rzeczywistych w łatwy sposób można wymusić ilość liczb wyświetlanych po przecinku: `printf("twoja liczba: %f, lub %.2f\n",a,a);`

Podstawową instrukcją sterującą w większości językach programowania jest instrukcja warunkowa `if`. Aby wykonały się instrukcje wewnątrz, musi zostać spełniony warunek. W przeciwnym wypadku mogą się wykonać instrukcje wewnątrz bloku `else`. Ważnym aspektem w trakcie programowania jest odpowiednie zagnieżdżanie instrukcji warunkowych.

```
1 #include <iostream>
2
3 int main()
4 {
5     float wzrost = 1.76;
6
7     if (wzrost < 1.6)
8         std::cout << "niski\n";
9     if (wzrost >= 1.6 && wzrost <=1.8)
10        std::cout << "sredni\n";
11     if (wzrost > 1.8)
12        std::cout << "wysoki\n";
13
14     if (wzrost < 1.6)
15        std::cout << "niski\n";
16     else if (wzrost > 1.8)
17        std::cout << "wysoku\n";
18     else
19        std::cout << "sredni\n";
20     return 0;
21 }
```

Inną istotną instrukcją warunkową jest instrukcja **switch**. Służy ona do podejmowania decyzji wyłącznie na podstawie wartości jednej zmiennej. Zmienną, na której chcemy pracować podajemy w nawiasach zaokrąglonych, które muszą znaleźć się zaraz po wystąpieniu słowa kluczowego **switch**. Do sprawdzenia wartości zmiennej podanej w instrukcji **switch** i wykonania danego fragmentu programu służy słowo kluczowe **case**. Z kolei słowo kluczowe **break** oznacza, że program powinien przerwać wykonywanie obecnej instrukcji sterującej, w tym przypadku switcha. W podanym przykładzie informuje ona kompilator, że dany case się wykonał i program ma już opuścić instrukcję sterującą switch. Jeśli zabraknie słowa kluczowego **break** to program wykona wszystkie instrukcje od momentu spełnienia warunku aż do samego końca switch'a. Instrukcja sterująca **switch** oferuje nam jeszcze jedną funkcjonalność, zapewnianą przez słowo kluczowe **default**. Wejście w kod znajdujący się po słowie kluczowym **default** nastąpi tylko wtedy, gdy żaden z case'ów nie zostanie spełniony.

```
1 #include <iostream>
2 int main()
3 {
4     int liczba;
5     std::cout << "Podaj liczbę: ";
6     std::cin >> liczba;
7     switch( liczba )
8     {
9         case 2:
10             std::cout << "dwa" << std::endl;
11             break;
12         case 1:
13             std::cout << "jeden" << std::endl;
14             break;
15         case 3:
16             std::cout << "trzy" << std::endl;
17             break;
18         default:
19             std::cout << "ani jeden, ani dwa, ani trzy" << std::endl;
20             break;
21     }
22     return 0;
23 }
```

2 Zadania

1. OCENA 3:

Napisz program, będący prostym kalkulatorem. Na początku spyta użytkownika o podanie dwóch liczb. Następnie podanie odpowiedniego operatora matematycznego (+, -, *, /). Na końcu wyświetl wynik. Pamiętaj, że nie dzielimy przez 0, więc należy w programie wyświetlić odpowiedni komunikat przy próbie wywołania operacji dzielenia przez 0. W przypadku wprowadzenia innego znaku niż przewidziane operatory matematyczne program powinien wyświetlić odpowiedni komunikat o nierozpoznaniu operacji.

2. OCENA 4:

Napisz program, który wyliczy miejsca zerowe równania kwadratowego w postaci

$$f(x) = ax^2 + bx + c. \quad (1)$$

Pamiętając, że:

- $a, b, c \in R$,
- $a \neq 0$,
- $\Delta = b^2 - 4ac$,
- dla $\Delta < 0$ nie ma miejsc zerowych,
- dla $\Delta = 0$ mamy jedno miejsce zerowe $x = \frac{-b}{2a}$,

- dla $\Delta > 0$ mamy dwa miejsca zerowe $x = \frac{-b+\sqrt{\Delta}}{2a}$ i $x = \frac{-b-\sqrt{\Delta}}{2a}$.

Funkcja pierwiastka z x : `sqrt(x)` z biblioteki `cmath`.

3. OCENA 5: W programie z zadania 2 należy uwzględnić następujące punkty:

- Obsługę równania liniowego, tożsamościowego i sprzecznego,
- zadana przez użytkownika precyzja wyświetlanych liczb (wykorzystaj odpowiednią funkcję z biblioteki *iomanip*),
- poprawne wypisanie wprowadzonego równania, np.: $x^2 - 2$ dla $a = 1, b = 0, c = -2$.

Uwaga

Wszystkie programy napisane podczas zajęć należy przesłać do prowadzącego drogą mailową (*Michał.Jaroszczuk@pwr.edu.pl*). Tytuł wiadomości musi być według następującego formatu `[skrótowa nazwa kursu] [termin] labX`, gdzie X to numer laboratorium. Należy przesłać jedynie pliki źródłowe (`*.cpp`), którą proszę skompresować do archiwum `nazwiskoX.zip`.

Tabela 1: grupy

[PP] [TP13]	labX	wtorek tydzień parzysty godz 13:15
[PP] [TP17]	labX	wtorek tydzień parzysty godz 17:05
[PP] [TN13]	labX	wtorek tydzień nieparzysty godz 13:15
[PP] [TN17]	labX	wtorek tydzień nieparzysty godz 17:05